

# COMPUTER SYSTEM AND ITS MAINTENANCE METHOD

Patent Number: JP2000276353

Publication date: 2000-10-06

Inventor(s): TERADA HIROSHI; YAMAZAKI HIROSHI

Applicant(s):: TOSHIBA CORP; TOSHIBA COMPUT ENG CORP

Requested Patent:  JP2000276353 (JP00276353)

Application Number: JP19990077196 19990323

Priority Number(s):

IPC Classification: G06F9/445 ; G06F3/06

EC Classification:

Equivalents:

---

## Abstract

---

**PROBLEM TO BE SOLVED:** To maintain a computer system from a USB FDD without depending upon a specific OS and a dedicated driver by equipping a program stored in a flash memory with a control processing for the USB FDD.

**SOLUTION:** The program stored in a boot block part of the flash BIOS-ROM 19 reads in a rewriting program from the USB FDD 21 or an FDD 17 and judges whether or not a read error occurs. When no error occurs, the control is transferred to the rewriting program read in from the FD, and data stored on the FD are read out to rewrite the data in the flash BIOS-ROM 19 or KBC 20. The rewriting program is read in by performing a read process from the FDD 17 and performing a read processing from the USB FDD 21 after the read processing from the FDD 17 ends in failure.

---

Data supplied from the [esp@cenet](mailto:esp@cenet) database - I2

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-276353

(P2000-276353A)

(43)公開日 平成12年10月6日 (2000.10.6)

(51)Int.Cl.<sup>7</sup>  
G 0 6 F 9/445  
3/06

識別記号  
3 0 1

F I  
G 0 6 F 9/06  
3/06

テ-マコ-ト(参考)  
4 2 0 L 5 B 0 6 5  
3 0 1 A 5 B 0 7 6

審査請求 未請求 請求項の数11 OL (全 12 頁)

(21)出願番号

特願平11-77196

(22)出願日

平成11年3月23日 (1999.3.23)

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(71)出願人 000221052

東芝コンピュータエンジニアリング株式会  
社

東京都青梅市新町3丁目3番地の1

(72)発明者 寺田 洋

東京都青梅市新町3丁目3番地の1 東芝  
コンピュータエンジニアリング株式会社内

(74)代理人 100083161

弁理士 外川 英明

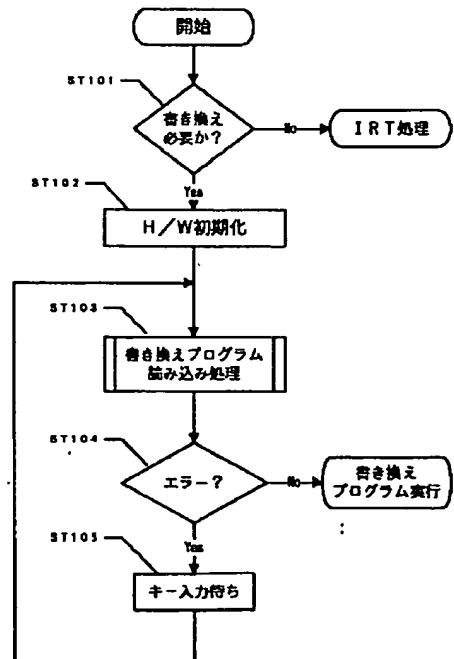
最終頁に続く

(54)【発明の名称】 コンピュータシステム及びそのメンテナンス方法

(57)【要約】

【課題】 USB FDDを利用して、コンピュータメンテナンスを実施する場合、USB FDDに対応した特定のOSと専用のドライバが必要であり、コンピュータシステム電源立ち上げ時、BIOS-ROMのプログラムで、USB FDDを制御して、コンピュータメンテナンスを実施することはできなかった。

【解決手段】 システムBIOS内にUSB FDDの制御処理ルーチン（メンテナンスプログラム）を持つことにより、コンピュータシステム電源立ち上げ時、特定のOSまたは専用のドライバに依存しない、USB FD Dからコンピュータシステムのメンテナンスを実行できる。



【特許請求の範囲】

【請求項1】 読み出し専用ブート領域にメンテナンスプログラムが格納されるフラッシュメモリから構成されたB IOS-ROMと、コンピュータシステムの電源立ち上げ時に、前記メンテナンスプログラムによって、前記フラッシュメモリの他の領域に対する書き換えの有無を判断し、前記フラッシュメモリの他の領域の書き換えが必要であると判断された場合には、前記コンピュータシステムの外部インターフェースであるユニバーサル・シリアルバス(USB)のホストコントローラを初期化し、前記USBに接続されたフロッピー・ディスク・ドライブにUSBアドレスを割当て、前記フロッピー・ディスク・ドライブの制御用データを前記ホストコントローラに設定し、前記ホストコントローラは、前記フロッピー・ディスク・ドライブ用制御データに基づき、前記フロッピー・ディスク・ドライブを制御し、前記フロッピー・ディスク・ドライブから読み取ったデータを前記BIOS-ROMに出力することを特徴とするコンピュータシステム。

【請求項2】 前記BIOS-ROMの読み出し専用ブート領域に格納された前記メンテナンスプログラムにより、前記USBに接続されたフロッピー・ディスク・ドライブに挿入されたフロッピー・ディスクからフラッシュメモリの書き換えデータと書き換え用プログラムを読み出し、その後、読み出された書き換え用プログラムに従い、前記フラッシュメモリに格納されていたデータを前記書き換えデータに書き換えることを特徴とする請求項1記載のコンピュータシステム。

【請求項3】 前記BIOS-ROMの読み出し専用ブート領域に格納された前記メンテナンスプログラムにより、前記フラッシュメモリの他の領域に格納された該コンピュータシステムの入出力ルーチンプログラムのエラー訂正の有無チェックを行い、エラー有りと判断された場合、前記フラッシュメモリの他の領域の書き換え要求を発生することを特徴とする請求項2記載のコンピュータシステム。

【請求項4】 ユーザからの要求に応じて、前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムにより、前記フラッシュメモリの他の領域の書き換え要求を発生することを特徴とする請求項2記載のコンピュータシステム。

【請求項5】 前記フロッピー・ディスク・ドライブに挿入されたフロッピー・ディスクに格納された書き換えプログラムの実行に伴い、前記書き換えデータを前記フラッシュメモリの他の領域に書き換えた後、コンピュータシステムのオペレーティングシステムを起動することを特徴とする請求項2記載のコンピュータシステム。

【請求項6】 前記BIOS-ROMの読み出し専用ブート領域に格納された前記メンテナンスプログラムは、所定の優先順位に従い、複数のフロッピー・ディスク・

ドライブから順次フロッピー・ディスク・ドライブの有無を判断することを特徴とする請求項1記載のコンピュータシステム。

【請求項7】 前記BIOS-ROMの読み出し専用ブート領域に格納された前記メンテナンスプログラムは、システム内部のフロッピー・ディスク・ドライブの有無を判断し、もしシステム内部のフロッピー・ディスク・ドライブが存在しないと判断された場合、USBに接続されたフロッピー・ディスク・ドライブの有無を判断することを特徴とする請求項6記載のコンピュータシステム。

【請求項8】 読み出し専用ブート領域にメンテナンスプログラムが格納されるフラッシュメモリから構成されたBIOS-ROMに於いて、コンピュータシステムの電源立ち上げ時に、前記メンテナンスプログラムによって、前記フラッシュメモリの他の領域に対する書き換えの有無を判断するステップと、前記フラッシュメモリの他の領域の書き換えが必要であると判断された場合には、前記コンピュータシステムの外部インターフェースであるユニバーサル・シリアルバス(USB)のホストコントローラを初期化するステップと、前記USBに接続されたフロッピー・ディスク・ドライブにUSBアドレスを割当てるステップと前記フロッピー・ディスク・ドライブの制御用データを前記ホストコントローラに設定するステップとを具備し、前記ホストコントローラは、設定されたフロッピー・ディスク・ドライブ用制御データに基づき、前記フロッピー・ディスク・ドライブを制御し、前記フロッピー・ディスク・ドライブから読み取ったデータを前記BIOS-ROMに出力することを特徴とするコンピュータシステムのメンテナンス方法。

【請求項9】 前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムにより、前記USBに接続されたフロッピー・ディスク・ドライブに挿入されたフロッピー・ディスクからフラッシュメモリの書き換えデータと書き換え用プログラムを読み出すステップと、その後、読み出された書き換え用プログラムの実行に従い、前記フラッシュメモリの他の領域を前記書き換えデータに書き換えることを特徴とする請求項8記載のコンピュータシステムのメンテナンス方法。

【請求項10】 前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムにより、前記フラッシュメモリの他の領域に格納された該コンピュータシステムの入出力ルーチンプログラムのエラー訂正の有無のチェックを行い、エラー有りと判断された場合、又は、ユーザからの要求に応じて、前記フラッシュメモリのメインブロック部の書き換え要求を発生するステップとを具備することを特徴とする請求項9記載のコンピュータシステムのメンテナンス方法。

【請求項11】 前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムによ

り、所定の優先順位に従い、複数のフロッピー・ディスク・ドライブから順次フロッピー・ディスク・ドライブの有無を判断することを特徴とする請求項10記載のコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、フラッシュメモリを使用したコンピュータシステムに係わり、特に、USB FDDを用いて、BIOS-ROM、又は、KBCのデータの内容を書き換える技術に関する。

【0002】

【従来の技術】従来、FDD (Floppy Disk Drive) を用いたコンピュータメンテナンスは、FDDに挿入されたメンテナンス用FDプログラムにより、メンテナンスに必要な情報をコンピュータシステムから読み込み、又は、コンピュータシステムに書き込みを行っていた。更に、メンテナンス用FDプログラムは、コンピュータシステムの診断用プログラムを具備し、コンピュータシステムの各回路が正常に動作するか診断を実行していた。

【0003】最近、コンピュータシステムのBIOS (Basic Input&Output System) -ROM (Read Only Memory) およびKBC (Key Board Controller) は、プログラムおよびデータを書き込むため、書き換え可能な不揮発性メモリ、又は、フラッシュメモリから構成され、プログラムおよびデータのバージョンアップおよび不良部分をメンテナンス用FDプログラムに格納されたプログラムおよびデータにより書き換えることが出来た。

【0004】特開平5-216639号公報には、BIOS-ROMをフラッシュメモリで構成し、コンピュータシステムの電源立上げ後、フラッシュメモリの書き込み禁止(読み出し専用ブート)領域に格納されたCRC (Cyclic Redundancy Check) プログラムを実行し、BIOS-ROMの内容にエラーが発見された場合、該ブートブロック部に格納された転送プログラムにより、BIOS-ROM内容の書き換え用FDに記憶されたBIOSファイルをフラッシュメモリに転送する技術を開示する。また、コンピュータシステムの電源立上げ時、キーボードの特定キーを押圧することにより、同様に、BIOS-ROM内容の書き換え用FDに記憶されたバージョンアップ用BIOSファイルをフラッシュメモリに転送する技術も開示する。

【0005】これら従来構成の内蔵FDDは、BIOS-ROMのプログラムがFDC (Floppy Disk Controller) と通信することによってデータの転送等を制御していた。従って、FDCの制御は、上位のアプリケーション(OSおよびドライバ)に依存することなく、BIOS-ROMがコンピュータシ

ステム電源立上げ時の初期化処理の段階で、FDDを用いて上記コンピュータの各種メンテナンスを実行可能であった。

【0006】近年、コンピュータシステムにおいては、周辺装置の拡張性の自由度を高めるため、USB (Universal Serial Bus) のサポートが始まられている。USBは、キーボード、マウス、カメラ等の様々なUSB対応デバイスとの通信に共通に使用できるシリアルインターフェース規格(1996年1月15日発行の「USB仕様書」Ver1.0)である。このUSBを用いることにより、周辺デバイス毎に用意されたコネクタをUSBコネクタに統合することができ、コンピュータシステムの低価格化及び拡張性の自由度を高めることが可能となる。

【0007】しかし、現状では、USBデバイスを使用できるようにするためには専用のドライバと特定のOS (Operating System)との組み合わせが必要である。この様なOS環境以外では、USBコネクタを持つコンピュータシステムにおいても、USBデバイスを使用することは出来ない。即ち、コンピュータシステム自身のHWがUSBコネクタをサポートしているにも係わらず、異なるOS環境では、USBコネクタを介してUSBデバイスを制御できなかった。

【0008】また、近年、ノートブックタイプのコンピュータシステムの薄型化が進み、FDDを内蔵する代りに、USBコネクタに接続するUSB FDDがサポートされるようになった。しかし、USB FDDを制御するには、上述の様に、専用のドライバと特定のOSが必要なため、従来のBIOS-ROMのプログラムでは、USB FDDを制御できない。従って、特定のOSと専用のドライバ無しでは、USB FDDを用いたコンピュータのメンテナンス、特に、フラッシュメモリから構成されるBIOS-ROMまたはKBCのデータを書き換えることは出来なかった。

【0009】

【発明が解決しようとする課題】上記した従来技術においては、USB FDDを利用して、コンピュータメンテナンスを実施する場合、USB FDDに対応した特定のOSと専用のドライバが必要であるという問題があった。

【0010】更に、従来技術においては、コンピュータシステムの電源立上げ時、BIOS-ROMのプログラムで、USB FDDを制御して、コンピュータメンテナンスを実施することはできなかった。

【0011】そこで、本発明は上記の問題を解決するためになされたものであり、フラッシュメモリに格納されたプログラムにより、USB FDDの制御を処理するため、特定のOSおよび専用のドライバに依然せず、コンピュータシステムのメンテナンスをUSB FDDから実施できる装置を提供することを目的とする。

【0012】

【課題を解決するための手段】この発明は、読み出し専用ブート領域にメンテナンスプログラムが格納されたフラッシュメモリから構成されたBIOS-ROMと、コンピュータシステムの電源立上げ時に、前記メンテナンスプログラムによって、前記フラッシュメモリの他の領域に対する書き換えの有無を判断し、前記フラッシュメモリの他の領域の書き換えが必要であると判断された場合には、前記コンピュータシステムの外部インタフェースであるユニバーサル・シリアルバス(USB)のホストコントローラを初期化し、前記ユニバーサル・シリアルバスに接続されたフロッピー・ディスク・ドライブにUSBアドレスを割当て、前記フロッピー・ディスク・ドライブの制御用データを前記ホストコントローラに設定し、前記ホストコントローラは、前記フロッピー・ディスク・ドライブ用制御データに基づき、前記フロッピー・ディスク・ドライブを制御し、前記フロッピー・ディスク・ドライブから読み取ったデータを前記BIOS-ROMに出力することを特徴とする。

【0013】このような構成によれば、コンピュータシステムの電源立上げ時、フラッシュメモリの読み出し専用ブート領域に格納されたメンテナンスプログラムにより、USB FDDの制御処理を有するため、特定のOSまたは専用のドライバに依然せず、コンピュータシステムのメンテナンスを実施することが可能となる。

【0014】また、この発明は、前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムは、所定の優先順位に従い、複数のフロッピー・ディスク・ドライブから順次フロッピー・ディスク・ドライブの有無を判断することを特徴とする。

【0015】更に、この発明は、前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムは、前記コンピュータシステム内部のフロッピー・ディスク・ドライブの有無を判断し、もし前記システム内部のフロッピー・ディスク・ドライブが存在しないと判断された場合、USBに接続されたフロッピー・ディスク・ドライブの有無を判断することを特徴とする。

【0016】このような構成によれば、前記BIOS-ROMの読み出し専用ブート領域に格納されたメンテナンスプログラムは、所定の優先順位に従い、内部FDDと、USB FDDのどちらか一方がコンピュータシステムにインプリメントされているか有無をチェックし、インプリメントされた方からコンピュータシステムのメンテナンスを実施することが可能となる。

【0017】

【発明の実施の形態】以下、図面を参照して本発明の実施形態を説明する。図1には、本発明の一実施形態に係わるコンピュータシステムの構成が示されている。このコンピュータシステムは、ノートブックタイプまたはサ

ブノートタイプのポータブルパーソナルコンピュータであり、コンピュータ本体と、このコンピュータ本体に開閉自在に取り付けられたLCDパネルユニットとから構成される。

【0018】コンピュータ本体には、PCIバス1、ISAバス2、CPUモジュール11、主メモリ12、VGAコントローラ13、ビデオメモリ(VRAM)14、PCIインターフェイスブリッジ(PCI I/F)15、I/Oコントローラ16、フロッピーディスクドライブ(FDD)17、ハードディスクドライブ(HDD)18、フラッシュBIOS-ROM19、キーボードコントローラ(KBC)20、USBデバイス(USB FDD)21などが設けられる。

【0019】CPUモジュール11は、このシステム全体の動作制御およびデータ処理を実行するものであり、ここにはCPU、キャッシュ、さらには主メモリ12を制御するためのメモリコントローラなどが搭載される。

【0020】主メモリ12は、このシステムの主記憶として使用されるものであり、オペレーティングシステム、処理対象のアプリケーションプログラム、およびアプリケーションプログラムによって生成されたユーザデータ等が格納される。

【0021】VGAコントローラ13は、このシステムのディスプレイモニタとして使用されるLCDまたは外部CRTを制御するためのものであり、VRAM14に描画された画面データをそれらディスプレイモニタに表示する。

【0022】PCIインターフェイスブリッジ(PCI I/F)15は、1チップLSIによって実現されたゲートアレイであり、ここには、PCIバス1とISAバス2との間を双方向で接続するブリッジ機能が内蔵されているほか、HDD18の制御を行うIDEコントローラなども内蔵される。

【0023】I/Oコントローラ16は、各種I/Oデバイスを制御するためのゲートアレイであり、PCMCIA/CARD BUS仕様のPCカードを制御する機能および、外部FDD接続端子とFDD接続ケーブルを介して、脱着可能な外部FDD17を制御するFDC機能の他、シリアルポート、パラレルポート、赤外線通信ポート、USB接続端子とUSB接続ケーブルを介して、脱着可能なUSBデバイス(USB FDD)を制御するUSB HC(Host Controller)などの制御機能を有する。

【0024】フラッシュBIOS-ROM19は、システムBIOS(Basic I/O System)を記憶するためのものであり、プログラム書き替えが可能ないようにフラッシュメモリによって構成される。システムBIOSは、このシステム内の各種ハードウェアをアクセスするためのファンクション実行ルーチンを体系化したものであり、ここには、USB FDDを制御するメ

ンテナンスする機能も含まれる。

【0025】また、フラッシュBIOS-ROM19には、システム環境設定を変更するためのセットアッププログラムも格納されており、所定のキー操作を行うことによりそのセットアッププログラムを起動することができる。あるいは、上記フラッシュBIOS-ROM19に格納されたUSB-FDD制御プログラムに従い、USB接続ケーブルを介して、USB-FDD21又は外部FDD17に挿入された外部FDによって供給されるセットアッププログラムを起動することもできる。

【0026】キーボードコントローラ(KBC)20は、キーボード(KB)、およびポインティングスティック(PS)またはマウスなどのポインティングデバイスの制御を行う。

【0027】次に、図2乃至図9のフローチャートを参照して、本発明の特徴とするシステムBIOS-ROM19、又は、KBC20のデータ書き換え時におけるシステムのメンテナンスについて説明する。

【0028】図2は、本実施形態におけるコンピュータシステム電源立上げ時のシステムBIOS-ROM19の書き換え禁止・読み出し専用領域(ブートブロック)部に格納された書き換える為のメンテナンス処理ルーチンであり、以下の通りに説明する。

【0029】電源スイッチのオンにより、コンピュータシステムが起動される。システムBIOS-ROM19のブートブロック部に格納されたプログラムは、コンピュータシステム起動に必要な最低限のハードウェアの初期化を実行後、システムBIOS-ROM19及び、又はKBC20内容のデータの書き換えが必要か否か判断する。即ち、KBの所定のファンクションキーの押圧によるユーザからの書き換え要求、又は、フラッシュメモリのメインブロックに記憶されたシステムBIOS自身のCRCエラーの検出に基づき、判断される(ST101)。

【0030】ST101の判断処理で、書き換えが必要無いと判断された場合(ST101のNo)、フラッシュBIOS-ROM19のメインブロックに記憶されたIRT(Initialization & Reliability Test)処理を実行し、コンピュータシステムの起動を続行する。

【0031】前記書き換えが必要かどうかの判断処理で、書き換えが必要であると判断された場合(ST101のYes)、システムBIOS-ROM19及び、又はKBC20のデータ書き換えのために、主メモリ12の初期化およびメッセージ表示のためのVGAコントローラの13初期化等を実施する(ST102)。

【0032】書き換えのために必要なハードウェア初期化が終了後、FDD17又はUSBFDD21から書き換えプログラムの読み込み処理ルーチンを行う(ST103)。この処理ルーチンの詳細は図3を用いて後述す

る。

【0033】フラッシュBIOS-ROM19のブートブロック部に記憶されたプログラムは、USB-FDD21又はFDD17から書き換えプログラムの読み込み処理を実施し、該読み込みのためのエラーが発生していないかどうか判断する(ST104)。

【0034】ST104の判断処理でエラーが発生していると判断された場合(ST104のYes)、書き換えプログラムが存在するFDをFDDに挿入する確認メッセージを表示して、KBからのキー入力待ち状態となる(ST105)。ST103の書き換えプログラムの読み込み処理は、まず第1に、FDD17から読み込み処理を実行し、FDD17からの読み込み処理に失敗後、USB-FDD21から読み込み処理を実行する。更に、USB-FDD17からの読み込み処理に失敗した場合、再度、FDD17から読み込み処理を実行し、順次この読み込み処理の手順を繰り返す。

【0035】ST104の判断処理でエラーが発生していないと判断された場合(ST104のNo)、FDから読み込まれた書き換えプログラムに制御を移し、FDに格納されたデータを読み出し、フラッシュBIOS-ROM19及び、又はKBC20のデータと書き換えを実行する。

【0036】次に、図3を参照して、本実施形態におけるUSB-FDD21からの書き換えプログラムの読み込み処理ルーチンを説明する。ST103の書き換えプログラムの読み込み処理ルーチンが実行されると、FD-D17から読み込み処理を試行する。FDD17と書き換え用FDがコンピュータシステムに接続されているか否か判断される(ST201)。

【0037】ST201の判断処理で、FDD17に書き換え用FDが挿入されていると判断された場合(ST201のYes)、I/Oコントローラ16にインプリメントされたFDCに所定のコマンドを送信し、該FDCのリセットを実行する(ST202)。

【0038】FDCのリセット処理終了後、FDタイプの識別処理で、ユーザにより挿入されたFDが、2DDなのか、2HDなのか、又はコンピュータシステムのサポート外のメディアなのかを判断する(ST203)。

【0039】前記FDタイプの識別処理で識別されたメディアタイプに従って、転送レートを設定し、設定された転送レートで、FDから書き換えプログラムを主メモリ12に読み込む(ST204～ST205)。

【0040】ST201の判断処理で、本コンピュータシステムが、FDD17が接続されていないと判断された場合(ST201のNo)、書き換えプログラム読み込み処理ルーチンは、次に、本コンピュータシステムがUSB-FDD21をサポートしているか否かを判断する(ST206)。

【0041】ST206の判断処理で、USB-FDD

21をサポートしていないと判断された場合(ST206のNo)、書き換えプログラム読み込み処理ルーチンは、ST105に制御を移す。

【0042】書き換えプログラム読み込み処理ルーチンは、図1の説明で述べたように、FDD17からの読み込み処理とUSB FDD21からの読み込み処理とを書き換えプログラムが正しく読み込めるまで交互に繰り返し実行される。

【0043】また、USB FDD21をサポートしていると判断された場合(ST206のYes)、USBをコントロールするためのUSB HCが初期化されているかどうかを判断する(ST207)。

【0044】このルーチンが初めて実行される場合、USB HCは初期化されていないので、USB HCの初期化処理を実行する(ST207のNo～ST208)。USB HCの初期化ルーチンの詳細は、図4を用いて後述する。

【0045】USB HCの初期化処理ルーチンの終了後、初期化でエラーが発生したかどうかを判断する(ST209)。ST209の判断処理で初期化エラーと判断された場合(ST209のYes)、初期化エラーとして書き換えプログラム読み込み処理ルーチンを終了する。

【0046】前記エラー判断処理でエラーなしと判断された場合(ST209のNo)、又は、USB FDD21からの書き換えプログラム読み込み処理ルーチンが実行されるのが2回目以降の場合、即ち、USB HCの初期化がすでに終了している場合(ST207のYes)、Test Unit Ready コマンドをUSB HCに送信し、USB FDD21のリセット処理を実行する(ST210)。コマンド送信処理ルーチンは、図8を用いて後述する。

【0047】ST210のUSB FDD21のリセット処理終了後、FDタイプの識別処理を実行し、USB FDD21に挿入されたFDが、2DDなのか、2HDなのか、またはサポート外のメディアなのかを判断する(ST211)。

【0048】FDタイプの識別が終了すると、USB FDDから書き換えプログラムを読み込むデータリード処理ルーチンを実行する(ST212)。このデータリード処理ルーチンの詳細は、図7を用いて後述する。

【0049】次に、図4を参照して、本実施形態におけるUSB HCの初期化処理ルーチンを説明する。USB HCの初期化処理ルーチンは、まず、割込み要求IRQ (Interrupt ReQuest) の禁止設定及びI/Oベースアドレスの設定、USBのPCIコンフィグレーションレジスタの初期化、USBフレームバッファが使用するメモリの初期化を実行する(ST301)。

【0050】ST301のUSB PCIコンフィグレ

ーションレジスタの初期化と使用するメモリの初期化終了後、USB HCにリセット信号を送信し、USB HCのリセットを実行する(ST302)。前記USB HCのリセット処理終了後、USB HCの動作を開始する(ST303)。

【0051】USB HCの動作開始後、USBコネクタを介して接続されたUSBデバイスの検出と接続処理ルーチンを実行する(ST304)。このUSBデバイスの検出と接続処理処理ルーチンの詳細は、図5を用いて後述する。

【0052】ST304のUSBデバイスの検出と接続処理ルーチンの終了後、初期化失敗判断処理で、前記USBデバイスの検出と接続処理ルーチン内でエラーが発生していないか否かを判断する(ST305)。

【0053】ST305の初期化失敗判断処理で、USBデバイスの検出と接続処理ルーチンが正常に終了したと判断された場合(ST305のNo)、USB HCの初期化は成功したとして、USB HCの初期化処理を終了する。

【0054】ST305の判断処理において、USBデバイスの検出と接続処理ルーチン内でエラーが発生し、USB HCの初期化に失敗したと判断された場合(ST305のYes)、USB HCの使用停止、USBポートの電源オフ、USBのI/Oアクセスの禁止、主メモリ12上のワークエリアのクリア等の切断処理を実行する(ST306)。

【0055】次に、図5を参照して、本実施形態におけるUSBデバイスの検出と接続処理ルーチンを説明する。USBデバイスの検出と接続処理ルーチンは、まず、USB HC内の所定レジスタを参照して、USBポートの接続ステータスをチェックする(ST401)。接続ステータスチェックにより、USBデバイスの接続状態を判断する(ST402)。接続ステータスチェック処理でUSBデバイスが接続されていないと判断された場合、USBデバイスの検出と接続処理ルーチンを終了する(ST402のNo)。

【0056】ST402の接続ステータスチェック処理でUSBデバイスが接続されていると判断された場合(ST402のYes)、USB HC内の所定レジスタのステータスチェンジビットをクリアする(ST403)。

【0057】ST402のYes～ST403の処理で、USBポートにUSBデバイスが接続されていることが確認後、USBポートのリセットおよび許可、USBデバイスの速度チェックなどのUSBデバイスのリセット処理を実行する(ST404)。

【0058】ST404のUSBデバイスリセット処理終了後、USBデバイスの種類判定と接続チェック処理ルーチンを実行する(ST405)。このUSBデバイスの種類判定と接続チェック処理ルーチンの詳細は、図

6を用いて後述する。

【0059】次に、図6を参照して、本実施形態におけるUSBデバイスの種類判定と接続チェック処理ルーチンを以下の通りに説明する。USBデバイスの種類判定と接続チェック処理ルーチンは、USBデバイスのディスクリピタを取得するため、USBデバイスに一回目のGet Descriptor Deviceコマンドを送信する(ST501)。

【0060】Get Descriptor Deviceコマンド送信後、アドレスがまだ割り当てられていないUSBデバイスが応答する。応答したUSBデバイスに対し、空きアドレスを一つ割り当てる(ST502)。USBデバイスの種類判定と接続チェック処理ルーチンは、アドレスが割り当てられたUSBデバイスに対し、二度目のGet Descriptor Deviceコマンドを送信する(ST503)。この二度目のGet Descriptor Deviceコマンドの送信により、USBデバイスのタイプ、例えば、USB FDDが確定する。

【0061】USBデバイスの種類判定と接続チェック処理ルーチンは、二度目のGet Descriptor Deviceコマンドの送信によって取得したデバイスディスクリピタから、コンフィグレーション可能かどうかを判断する(ST504)。

【0062】ST504の判断処理で、コンフィグレーション不可能であると判断された場合(ST504のNo)、USBデバイスの接続処理は失敗したと判断して、USBデバイスのタイプのクリアおよびUSBポートの電源オフなどの切断処理を実行し、エラーとして本処理を終了する(ST505)。

【0063】ST504の判断処理によって、コンフィグレーション可能であると判断された場合(ST504のYes)、USBデバイスに対しGet Descriptor Configurationコマンドを送信する(ST506)。このGet Descriptor Configurationコマンドの送信により取得した情報に基づき、USBデバイスに対しSet Descriptor Configurationコマンドの送信を実行する(ST507)。

【0064】USBデバイスによっては一つのデバイスに複数のインターフェースを備えているものもあるため、インターフェースの数だけUSBデバイスの詳細チェックを実行する(ST508)。この詳細チェックによってUSB FDDが存在しないと判断された場合、エラーとして本処理ルーチンを終了する。

【0065】次に、図7を参照して、本実施形態におけるUSB FDDからのデータリード処理ルーチンを以下の通りに説明する。USB FDDからのデータリード処理ルーチンは、一連のStart Stop Unitコマンドの送信、割り込みデータパケット受信と、Request Senseコマンドの送信により、USB FDD21のモータをオンにする(ST601～ST603)。

【0066】USB FDD21のモータをオン処理後、FDのメディアタイプが確定されているか否か判断する(ST604)。このメディアタイプ判断処理により、メディアタイプが確定されていないと判断された場合(ST604のNo)、USB FDD21に対しRead Format CapacitiesコマンドとMode Senseコマンドを送信する(ST605～ST606)。FDのステータス取得後、FDのメディアタイプを確定する(ST607)。

【0067】前記メディアタイプ判断処理により、メディアタイプが確定されていると判断された場合(ST604のYes)、及びST605～ST607の処理によりFDのメディアタイプが確定された場合、FDのヘッド数、シリング数、セクタ数、ブロック数を設定する(ST608)。

【0068】FDのヘッド数、シリング数、セクタ数、ブロック数の設定終了後、これらの各設定値が、確定したFDのメディアタイプに対して正しいか否かを判断する(ST609)。ST609の判断処理によって正しくないと判断された場合(ST609のNo)、エラーとして本処理ルーチンを終了する。

【0069】設定したヘッド数、シリング数、セクタ数、ブロック数の各設定値が、確定したFDのメディアタイプに対して正しいと判断された場合(ST609のYes)、転送バッファとLBA(Logical Block Address)を設定し、バルク転送によってFDDからデータを受信する(ST610～ST612)。

【0070】USB FDD21からデータ受信終了後、割り込みデータパケット受信により、終了ステータスを取得し、本処理ルーチンを終了する(ST613)。次に、図8と図9を参照して、本実施形態におけるシステムBIOSのプログラムからUSB HCに対するコマンド送信処理と設定コマンド送信の処理ルーチンを以下の通りに説明する。

【0071】USB HCに対するコマンドの送信処理ルーチンは、通常コントロール転送によって実行される。コマンド送信の方法は、四種類(コントロール、割り込み、バルク、アイソクロノス)存在し、送信するコマンドの種類によって送信方法が決定される。即ち、USB HCに対するコマンド送信処理ルーチンは、図8のST701～ST709迄の処理に図9のST901～ST905の処理を追加した処理と、図9の単独のST901～ST905の処理がある。

【0072】図3記載のST210の「Test Unit Readyコマンド」、図7記載のST601の「Start Stop Unitコマンド」、図7記載のST605の「Read Format Capacitiesコマンド」、図7記載のST606の「Mode Senseコマンド」は、図8記載のST701から開始されるコマンドである。

【0073】図6記載のST501、ST503の「Get Descriptor Device コマンド」、図6記載のST506の「Get Descriptor Configurationコマンド」、図6記載のST507の「Set Descriptor Configurationコマンド」、図7記載のST603と図8記載のST704の「Request Sense コマンド」と図8記載のST703の「Clear Stall Request コマンド」は、図9記載のST801から開始されるコマンドである。

【0074】なお、「Request Sense コマンド」は、コントロール転送でコマンドを送信後、バルク転送により図9記載のST801～ST805の処理をもう一度行い、データを受信する。

【0075】図8記載のST701から開始される処理は、ST701で、図9記載のST801～ST805の処理を実行する。まず、図9記載のST801から開始される設定コマンド送信処理ルーチンについて以下の通り説明する。

【0076】USB HCに対するコマンド送信処理は、まず、コマンド実行に必要となるTDの個数を計算して求める(ST801)。続いて、リクエストデータの設定、QHの設定、TDの設定を行い(ST802～ST804)、TDを実行して(ST805)、本処理ルーチンを終了する。

【0077】次に、図8記載のST701から開始されるコマンド送信処理ルーチンについて、以下の通りに説明する。USB HCに対するコマンド送信処理ルーチンは、設定されたコマンドを送信するため、図9記載のST801～ST805の処理ルーチンを実行する(ST701)。

【0078】TD実行終了後、本コマンド送信処理ルーチンは、TD実行エラーかどうか判断する(ST702)。ST702の判断処理でエラーが無いと判断された場合、本処理ルーチンを終了する(ST702のNo)。前記判断処理でエラーと判断された場合(ST702のYes)、Clear Stall Request コマンドとRequest Sense コマンドを送信して、USB FDD21からステータスを取得する(ST703～ST704)。

【0079】USB FDD21からステータス取得後、Request Sense コマンドの送信結果がエラーか否か判断する(ST705)。前記判断処理でエラーが無いと判断された場合には(ST705のNo)、USB FDD21に対し再度TD実行を試行する。

【0080】前記Request Sense コマンド送信エラー判断処理でエラーと判断された場合(ST705のYes)、エラー内容がUSB FDCエラーであるか否かを判断する(ST706)。この判断処理で、USB FDCエラーではないと判断された場合(ST706のNo)、再度TD実行を試みる。

【0081】前記エラー内容判断処理でUSB FDCエラーと判断された場合(ST706のYes)、US

B FDD21が正しく接続できていないと判断する。例えば、コマンド送信の途中などでUSB FDD21がはずされてしまった場合などにUSB FDCエラーとなる場合がある。USB FDCエラーの場合、USB FDD21を再認識するため、USBポートのリセットおよび許可、デバイスの速度チェックなどを実行するUSBデバイスリセット処理、デバイスの種類判定と接続チェック処理(図6)、USB FDDデバイス情報データの取得を実行し、再度コマンド送信を試行する(ST707～ST709)。再度コマンド送信を試みてエラーになった場合、コマンドの送信は失敗と見なす。

【0082】尚、本願発明の実施形態のUSB FDD21では、USBデバイスコネクタに共通規格のUSBケーブルを介して接続できるため、パーソナルコンピュータの機種によって、外部FDD接続端子の形状が異なり、外部FDDをパーソナルコンピュータ本体に接続できないというような不具合がなくなる。

【0083】また、本願発明の実施形態では、従来、コンピュータシステムのメンテナンスをFDD17で実行していたが、USB FDD21で実行できるので、既存のFDCおよび外部FDD接続端子を取り除いたコンピュータシステムの構成が可能となる。従って、近年のコンピュータシステムに求められる小型化、軽量化、コストダウンに貢献することが出来る。

#### 【0084】

【発明の効果】以上説明したように、この発明によれば、フラッシュメモリに格納されたプログラムにUSB FDDの制御処理を具備することにより、特定のOSおよび専用のドライバに依然せず、コンピュータシステムのメンテナンスをUSB FDDから実施できる。

#### 【図面の簡単な説明】

【図1】本発明の一実施形態に係わるコンピュータシステムのシステム構成を示すブロック図。

【図2】本実施形態におけるコンピュータシステム電源立ち上げ時のシステムBIOS-ROM19のブートプロックに格納された書き換える為のメンテナンス処理ルーチンを示すフローチャート。

【図3】本実施形態におけるUSB FDD21からの書き換えプログラムの読み込み処理ルーチンを示すフローチャート。

【図4】本実施形態におけるUSB HCの初期化処理ルーチンを示すフローチャート。

【図5】本実施形態におけるUSBデバイスの検出と接続処理ルーチンを示すフローチャート。

【図6】本実施形態におけるUSBデバイスの種類判定と接続チェック処理ルーチンを示すフローチャート。

【図7】本実施形態におけるUSB FDDからのデータリード処理ルーチンを示すフローチャート。

【図8】本実施形態におけるUSB FDDを制御する

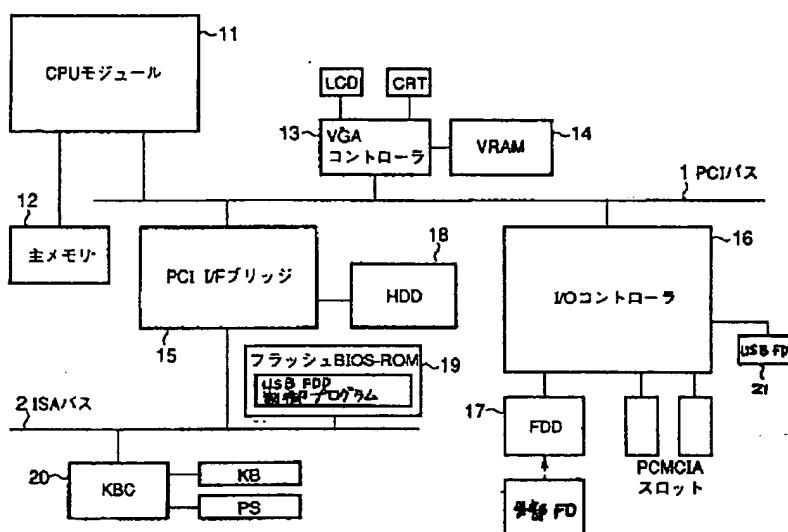
ためにシステムB IOSのプログラムがUSB HCに対するコマンド送信処理ルーチンを示すフローチャート。

【図9】本実施形態におけるUSB FDDを制御するための設定コマンド送信処理ルーチンを示すフローチャート。

【符号の説明】

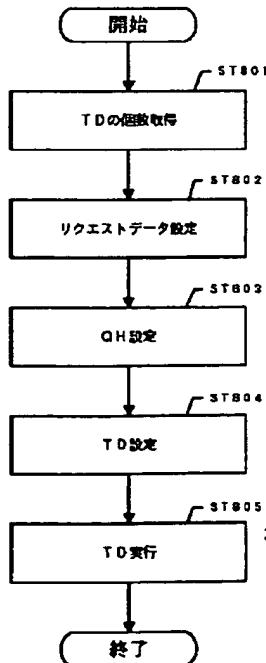
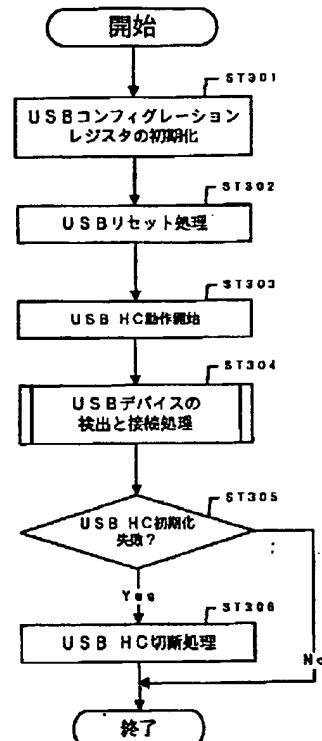
11…CPUモジュール、12…主メモリ、13…VGA  
Aコントローラ、14…VRAM、15…PCIインターフェースブリッジ、16…I/Oコントローラ、17…外部FDD、19…フラッシュBIOS-ROM、21…USB FDD

【図1】

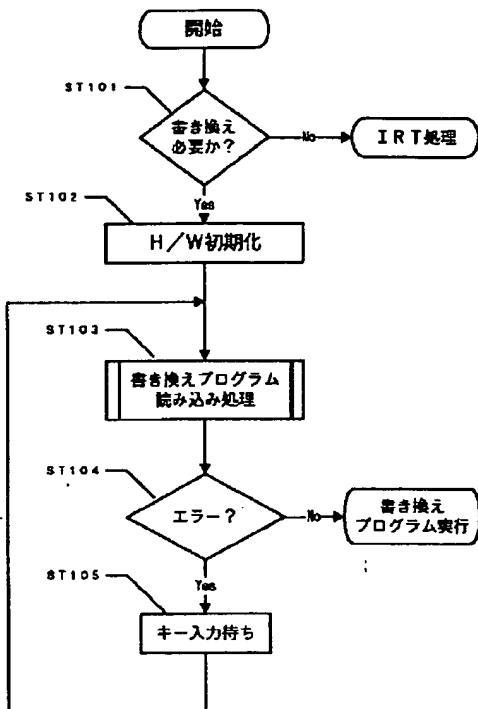


【図9】

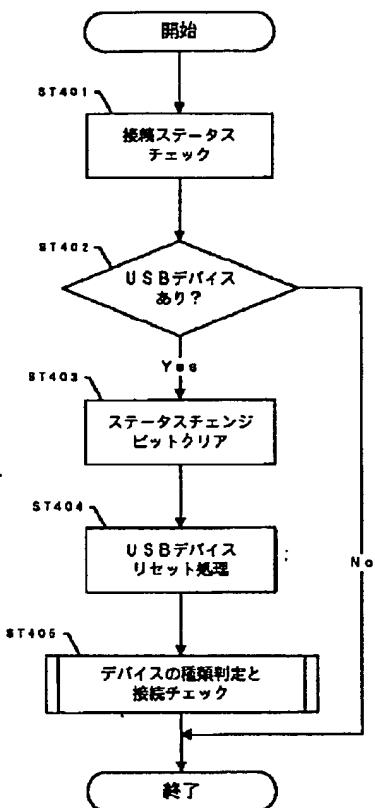
【図4】



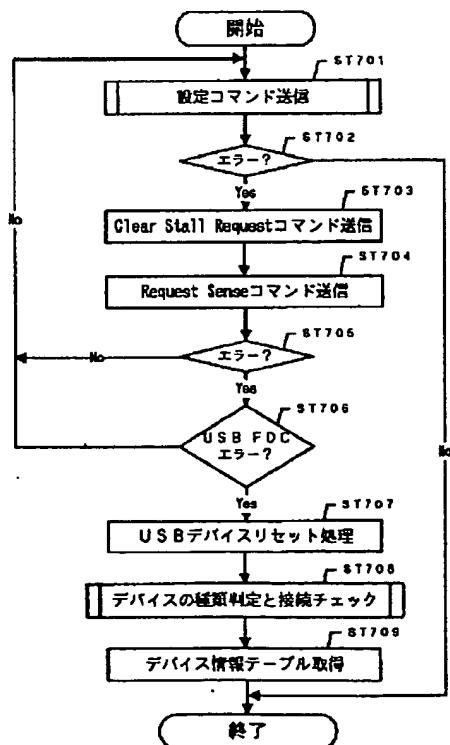
【図2】



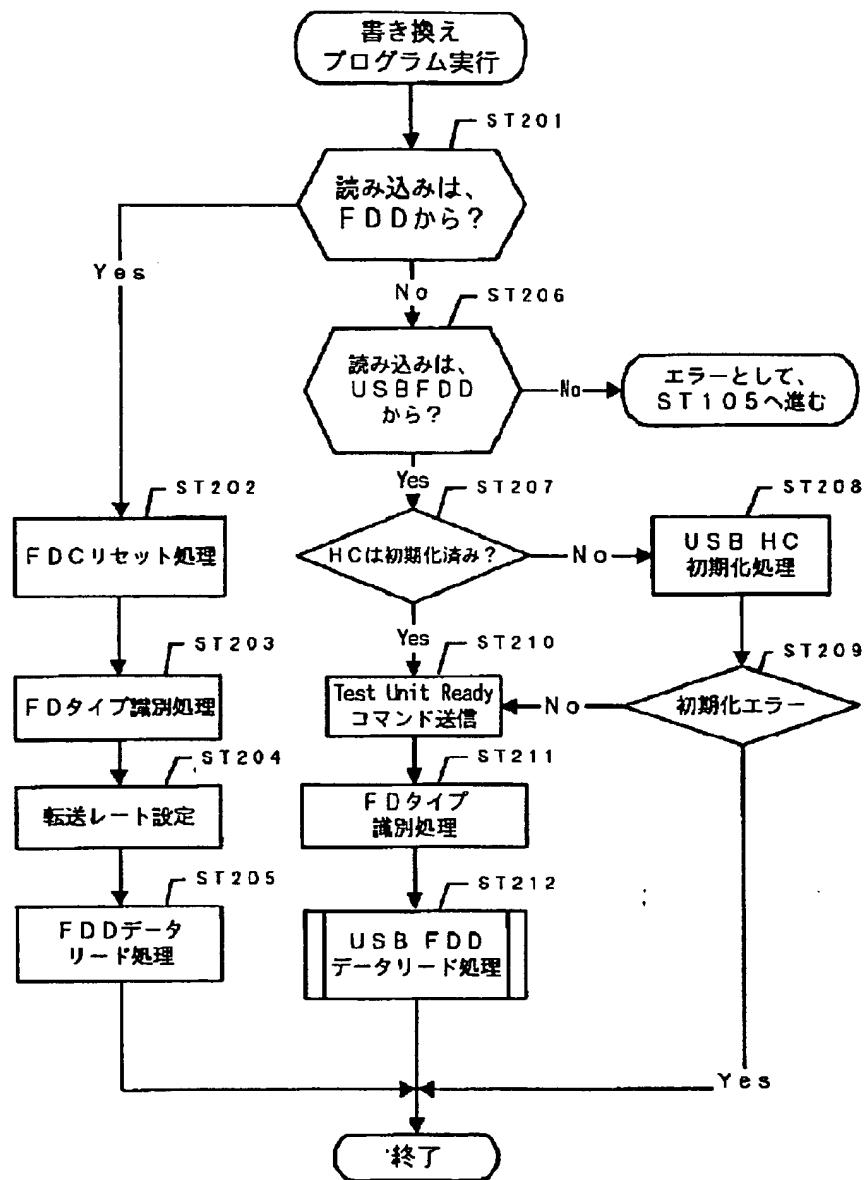
【図5】



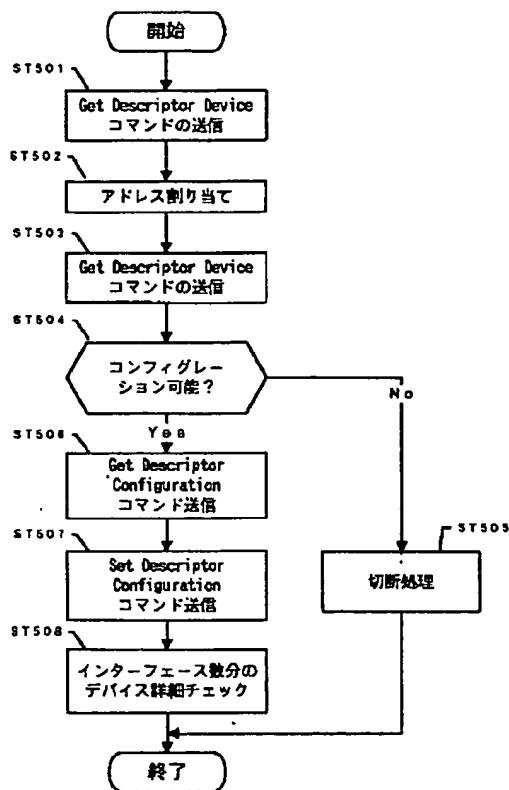
【図8】



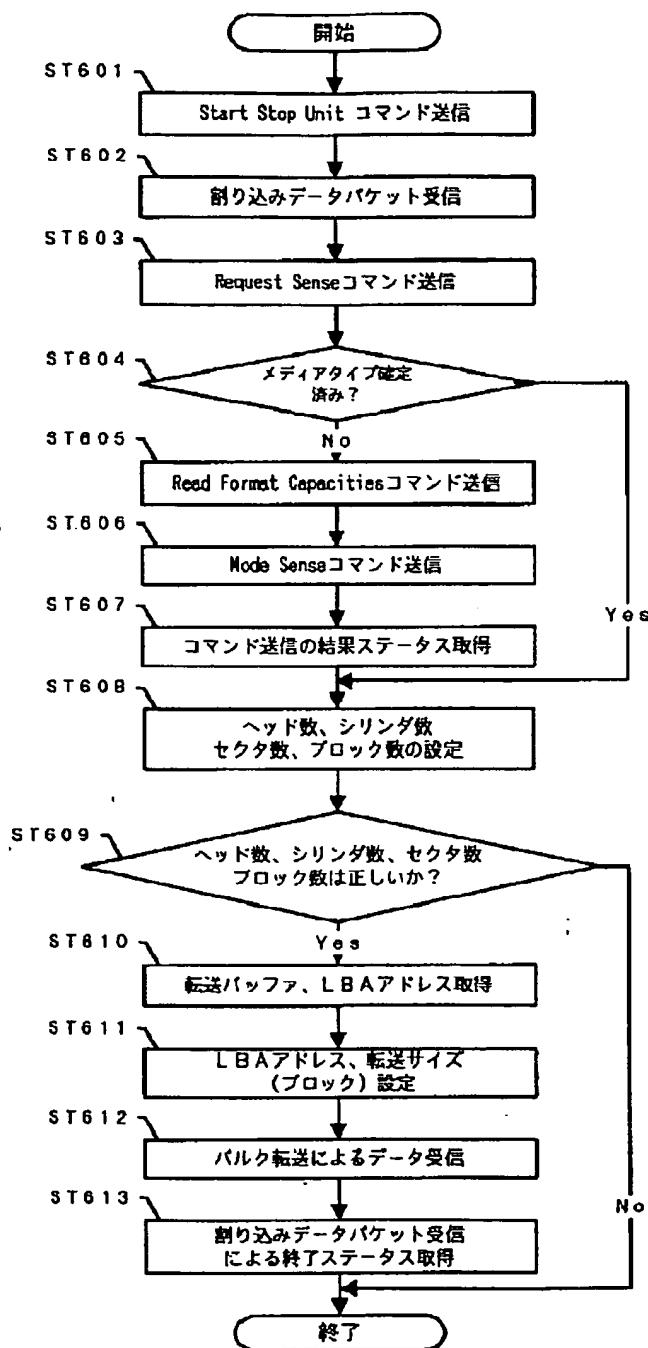
【図3】



【図6】



【図7】



フロントページの続き

(72)発明者 山崎 浩  
東京都青梅市新町3丁目3番地の1 東芝  
コンピュータエンジニアリング株式会社内

Fターム(参考) 5B065 BA02 BA05 CA01 ZA05  
5B076 BB18

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**